



# Customizable Collaborative Modeling for Domain-specific Modeling Languages

Philip Langer

planger@eclipsesource.com

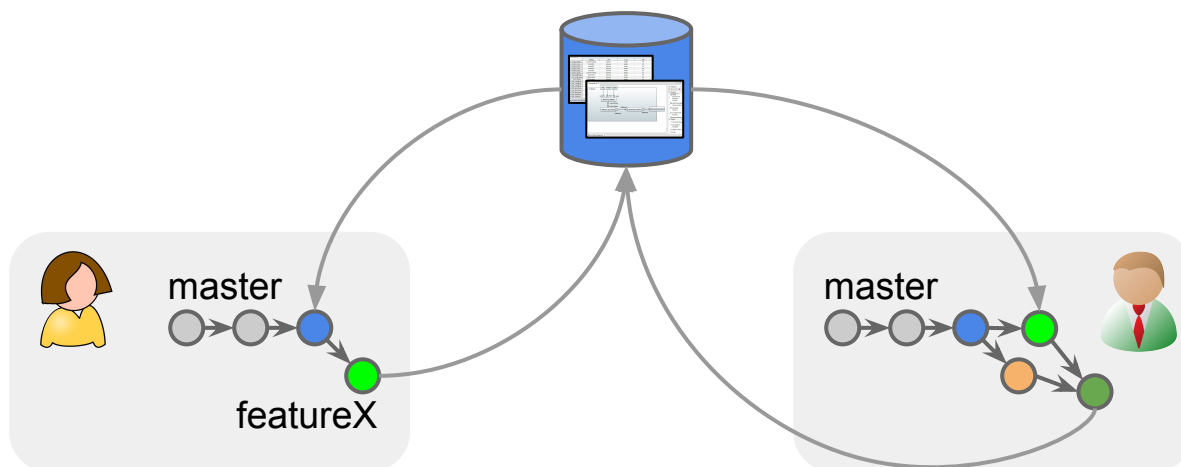
Maximilian Koegel

mkoegel@eclipsesource.com

## Collaborative Modeling (“Version Control for Models”)

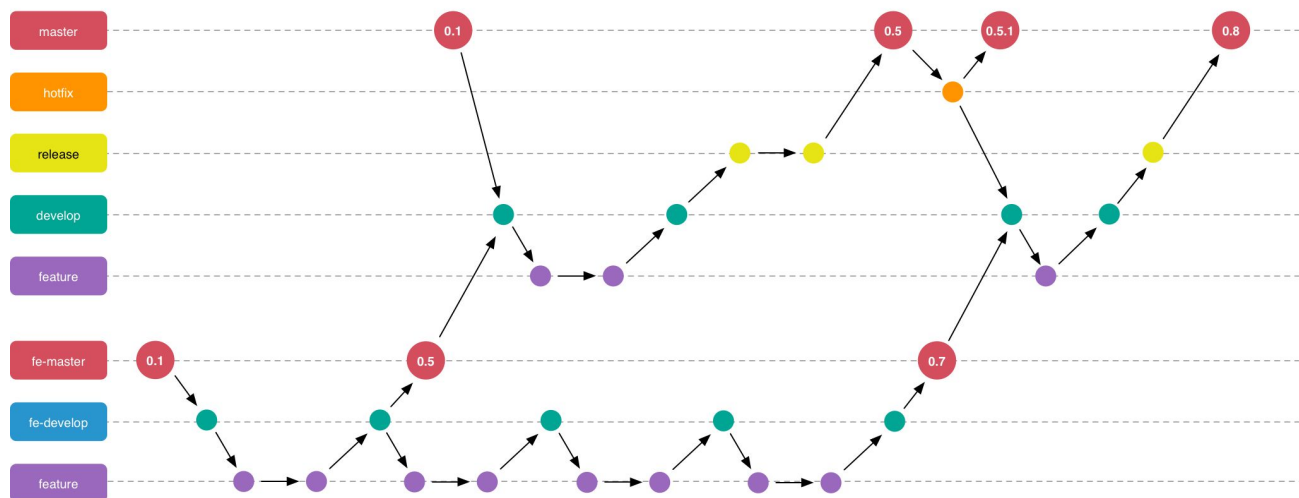


- Processes and tooling that enable...
  - working on **shared models**
  - changing models separately in **isolated branches**
  - obtaining **differences** among model versions
  - **sharing changes** with others
  - **merging changes** with changes others have made



## Collaborative Modeling

- Processes and tooling that enable to efficiently use **git for models**

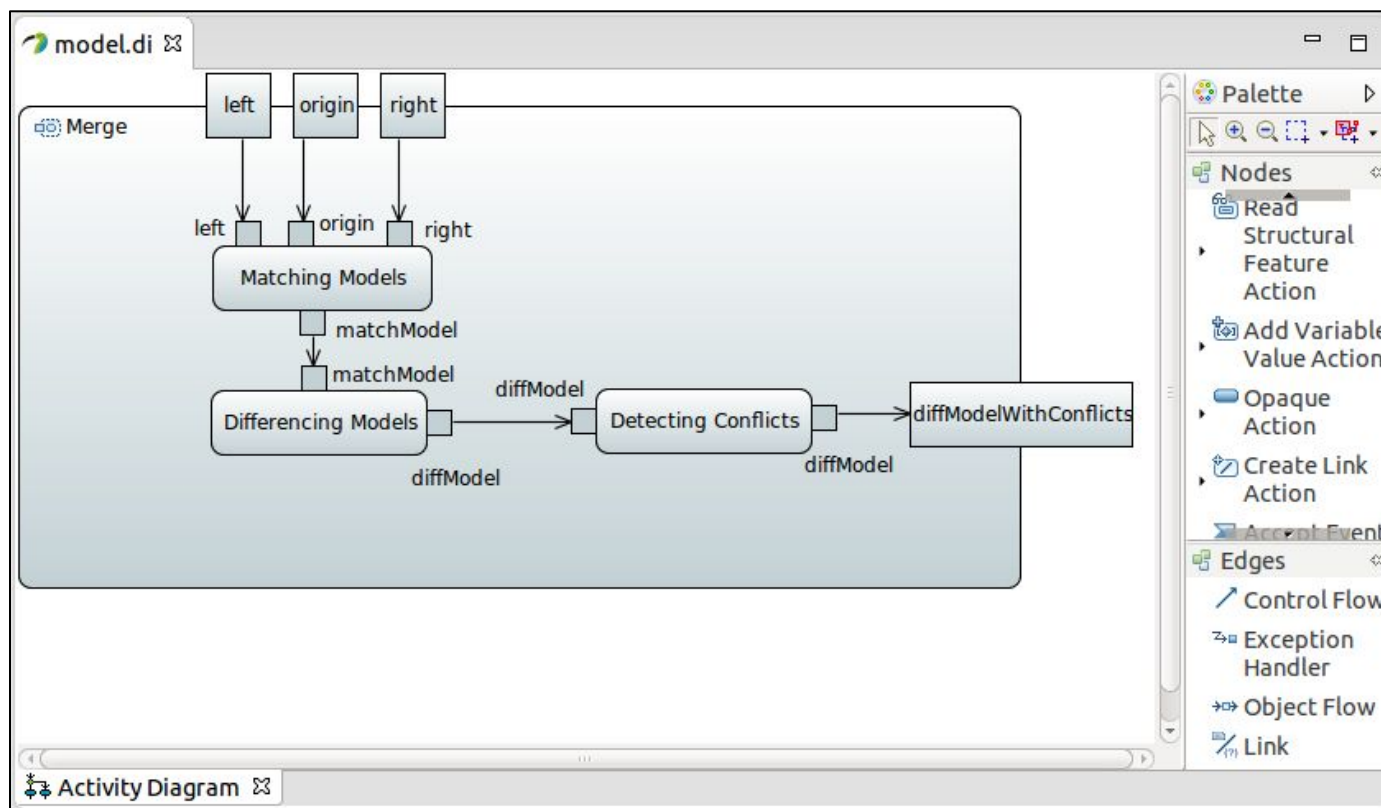


Taken from <https://github.com/KnowitLabs/Project-Workflow-and-Conventions>

- Isn't git on its own enough?

## Collaborative Modeling

- Processes and tooling that enable efficient use of **git for models**



## Collaborative Modeling

- Processes and tooling that enable to efficiently use **git** for models

<pre> 1&lt;?xml version="1.0" encoding="UTF-8"?&gt; 2&lt;uml:Model xmi:version="20131001" xmlns:xmi="http://www.omg.org 3  &lt;packageImport xmi:type="uml:PackageImport" xmi:id="_86TG0RaC 4    &lt;importPackage xmi:type="uml:Model" href="pathmap://UML_L 5  &lt;/packageImport&gt; 6  &lt;packagedElement xmi:type="uml:Activity" xmi:id=" 890fABaCEeW 7    &lt;edge xmi:type="uml:ControlFlow" xmi:id=" _BTR5QBaDEeWU9uWY 8    &lt;edge xmi:type="uml:ControlFlow" xmi:id=" _0Ji9MGXyEeWNf60-b 9    &lt;edge xmi:type="uml:ControlFlow" xmi:id=" 1CbF8GXyEeWNf60-b 10   &lt;edge xmi:type="uml:ControlFlow" xmi:id=" q7HXYGX1EeWNf60-b 11   &lt;node xmi:type="uml:InitialNode" xmi:id=" _meUwBaCEeWU9uWY 12   &lt;node xmi:type="uml:OpaqueAction" xmi:id=" _APfhABaDEeWU9uWY 13     &lt;language&gt;Natural language&lt;/language&gt; 14     &lt;language&gt;JAVA&lt;/language&gt; 15   &lt;body&gt;Get the logical structure from the metamodel&lt;/body&gt; 16   &lt;body&gt;System.out.println(&amp;quot;whaat&amp;quot;);&amp;#xD; 17 s&amp;#xD; 18 asdfklkj&lt;/body&gt; 19   &lt;/node&gt; 20   &lt;node xmi:type="uml:OpaqueAction" xmi:id=" otLagGXyEeWNf60- 21   &lt;node xmi:type="uml:ActivityFinalNode" xmi:id=" vi o0GXyEeW 22   &lt;node xmi:type="uml:OpaqueAction" xmi:id=" mQZuSGX1EeWNf60- </pre>	<pre> 1&lt;?xml version="1.0" encoding="UTF-8"?&gt; 2&lt;uml:Model xmi:version="20131001" xmlns:xmi="http://www.omg. 3  &lt;packageImport xmi:type="uml:PackageImport" xmi:id="_86TG0 4    &lt;importPackage xmi:type="uml:Model" href="pathmap://UM 5  &lt;/packageImport&gt; 6  &lt;packagedElement xmi:type="uml:Activity" xmi:id=" 890fABaC 7    &lt;edge xmi:type="uml:ControlFlow" xmi:id=" _BTR5QBaDEeWU9u 8    &lt;edge xmi:type="uml:ControlFlow" xmi:id=" _0Ji9MGXyEeWNf6 9    &lt;edge xmi:type="uml:ControlFlow" xmi:id=" 1CbF8GXyEeWNf6 10   &lt;node xmi:type="uml:InitialNode" xmi:id=" _meUwBaCEeWU9u 11   &lt;node xmi:type="uml:OpaqueAction" xmi:id=" _APfhABaDEeWU9 12     &lt;language&gt;Natural language&lt;/language&gt; 13     &lt;language&gt;JAVA&lt;/language&gt; 14   &lt;body&gt;What not&lt;/body&gt; 15   &lt;body&gt;System.out.println(&amp;quot;whaat&amp;quot;);&amp;#xD; 16 s&amp;#xD; 17 asdfklkj&lt;/body&gt; 18   &lt;/node&gt; 19   &lt;node xmi:type="uml:OpaqueAction" xmi:id=" otLagGXyEeWNf 20   &lt;node xmi:type="uml:ActivityFinalNode" xmi:id=" vi o0GXy 21   &lt;/packagedElement&gt; 22 &lt;/uml:Model&gt; </pre>
---	--

## Collaborative Modeling

- Processes and tooling that enable to efficiently use **git for models**
- **On model-level**
- **For Papyrus models**

→ **Seamless integration of**



## Model Versioning with EMF Compare and EGit

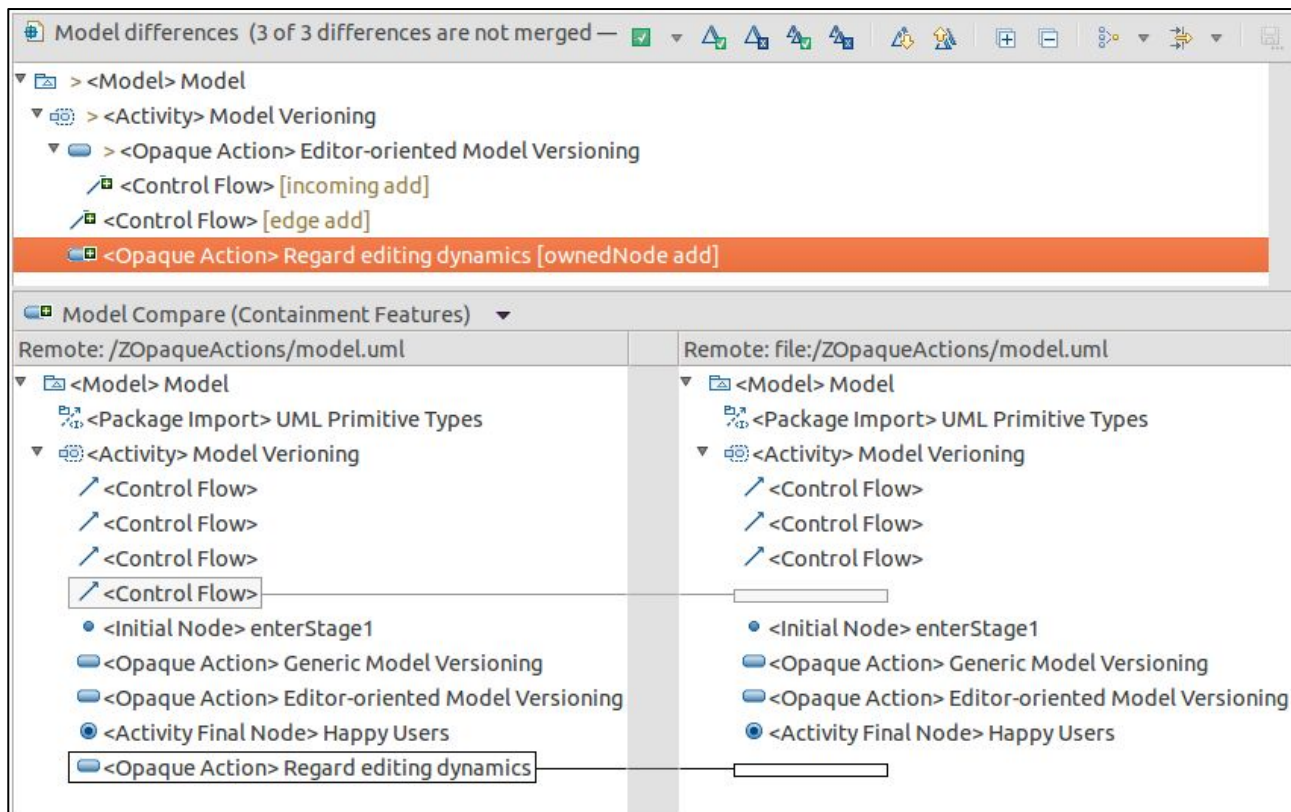
- EMF Compare

- A generic model versioning framework
- Raise the unit of versioning from *text line* to *model elements*
- Compare and merge the *logical structure* of the model
- *Generic* algorithms
  - Make use of the metamodel to obtain knowledge on logical structure
  - Reflection API to access and compare values generically
  - Support every modeling language that is specified in EMF

- ✓ Model-level comparison
- ✓ Model-level conflict detection
- ✓ Model-level merging



## Model Versioning with EMF Compare and EGit



This is already much closer to the modeling level!

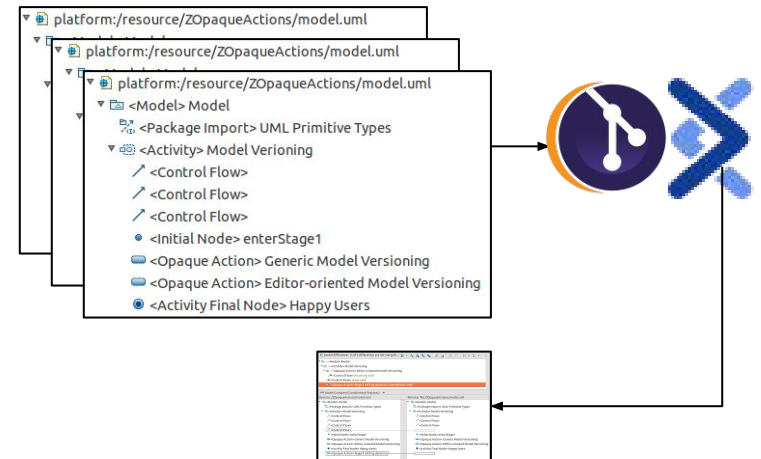


## Model Versioning with EMF Compare and EGit

- EMF Compare and EGit (integration of git in Eclipse)
  - EMF Compare extends EGit handling the involved model resources in
    - comparisons among branches, references, tags
    - git merge, rebase, and cherry-pick
  - EMF Compare handles not only single model resources
    - Git has a file-by-file way of working
    - Models often span across multiple files
    - “Logical models” are treated as a whole
    - Support for model fragmentation operations
- ✓ Model-level comparison and merging in all git operations
- ✓ Fix for paradigm mismatch: file vs logical model

# Gap between Model Versioning Tools and Modeling Editors

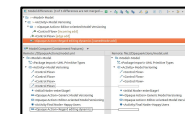
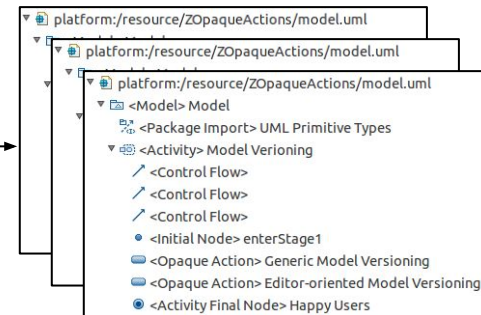
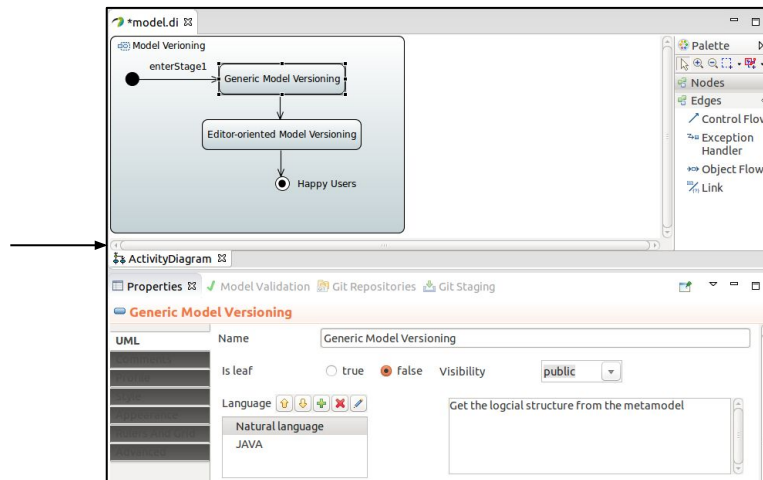
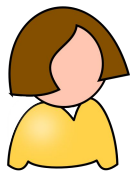
This is already much closer to the modeling level!



# Gap between Model Versioning Tools and Modeling Editors

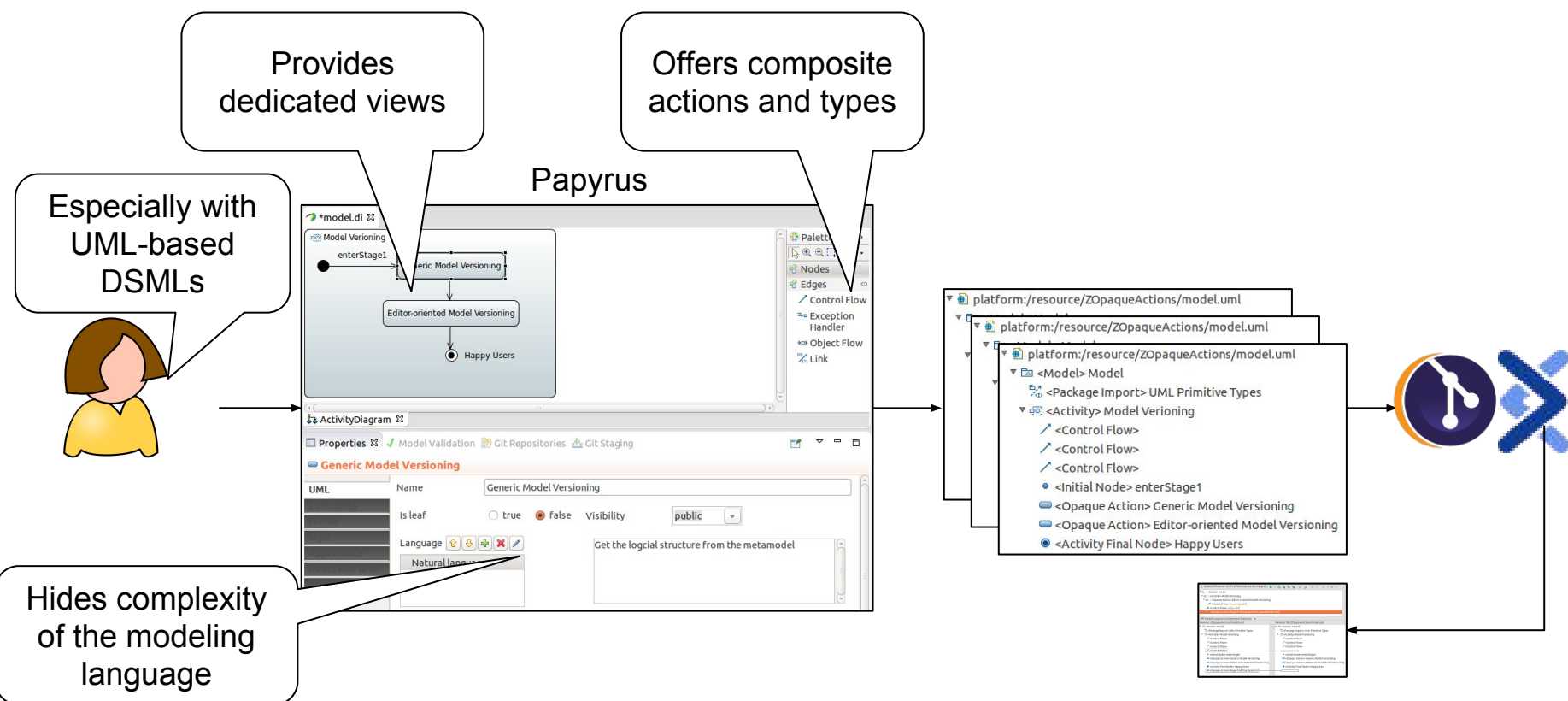
But this is not close enough!

## Papyrus



# Gap between Model Versioning Tools and Modeling Editors

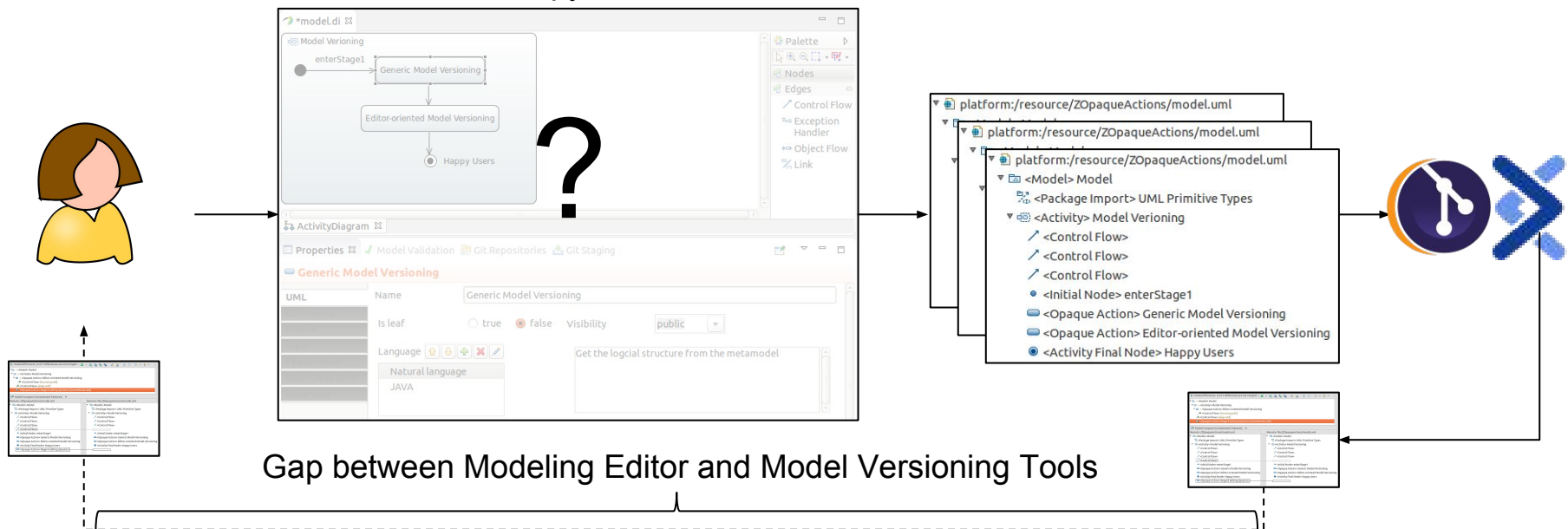
But this is not close enough!



# Gap between Model Versioning Tools and Modeling Editors

But this is not close enough!

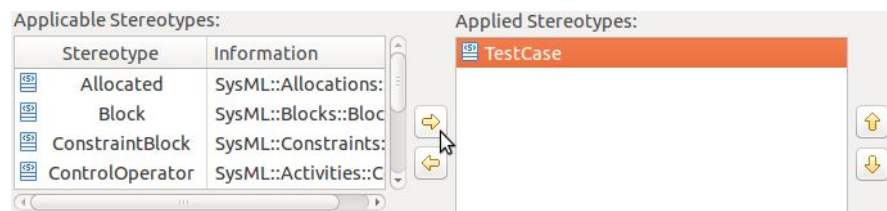
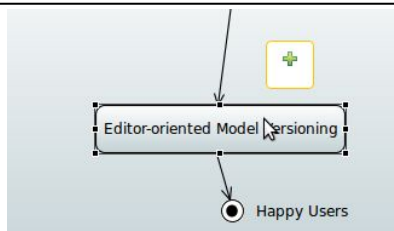
Papyrus



# Gap between Model Versioning Tools and Modeling Editors

## Papyrus

How a user applies a change



## EMF Compare

How EMF Compare sees the change

```

▼ □ > Shape <Opaque Action> Editor-oriented Model Versioning
  ▼ ◆ > Bounds 112
    ■ 112 [x changed]
    ■ 114 [y changed]
  
```

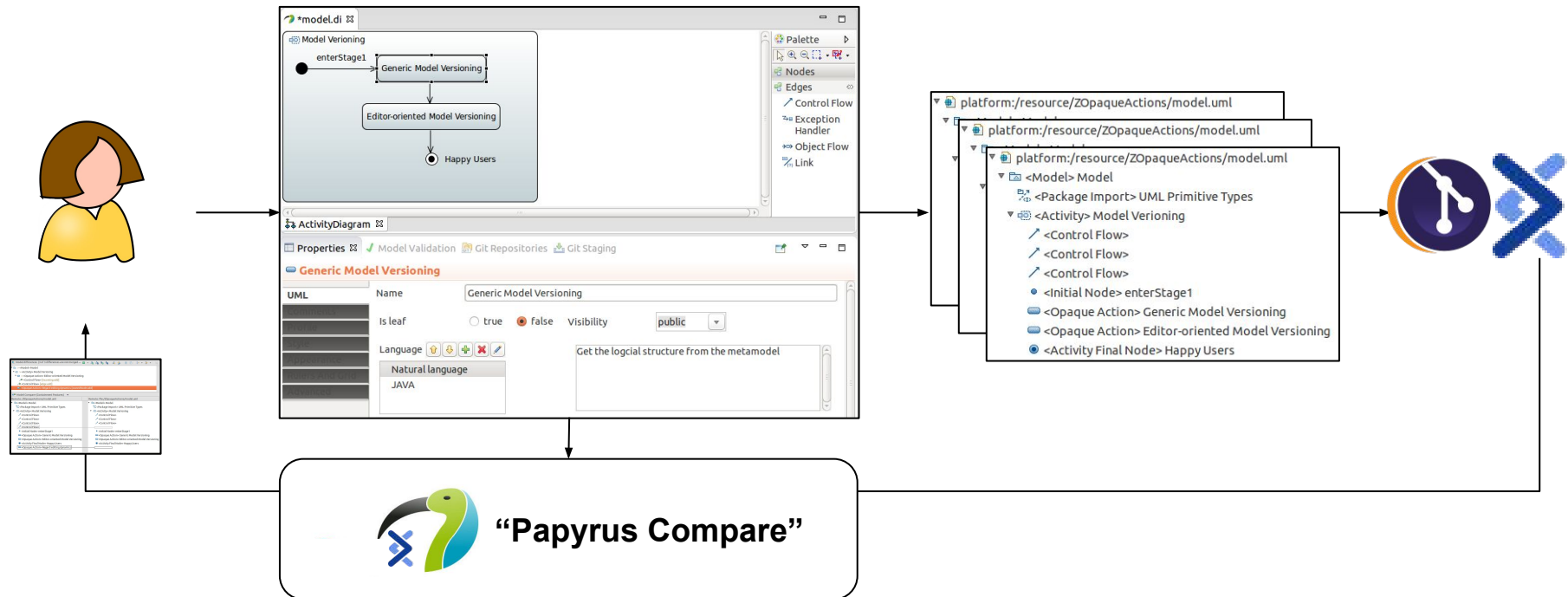
```

▼ 📁 platform:/resource/ZOpaqueActions/sysml.uml <-> file:/ZOpaqueActions/sysml.uml
  📄 Test Case [controlled in platform:/resource/ZOpaqueActions/sysml.uml]
  ▼ 📄 > Test Case
    📄 <TestCase> Activity1 [base_Behavior set]
  
```

...

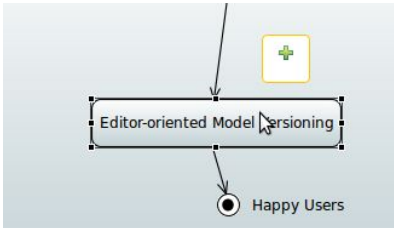

# Gap between Model Versioning Tools and Modeling Editors

## Papyrus





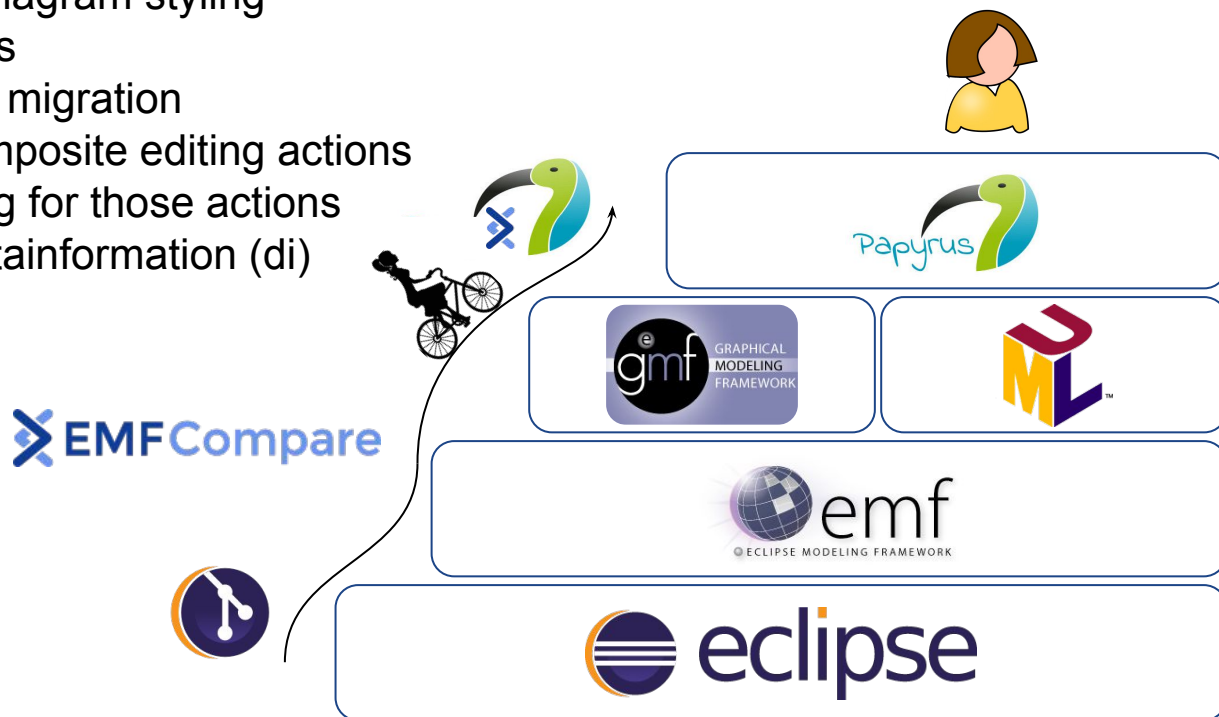
# Gap between Model Versioning Tools and Modeling Editors

Papyrus	EMF “Papyrus Compare”										
How a user applies a change	How Papyrus Compare shows the change										
											
<p>Applicable Stereotypes:</p> <table border="1"> <thead> <tr> <th>Stereotype</th><th>Information</th></tr> </thead> <tbody> <tr> <td>Allocated</td><td>SysML::Allocations::</td></tr> <tr> <td>Block</td><td>SysML::Blocks::Bloc</td></tr> <tr> <td>ConstraintBlock</td><td>SysML::Constraints::</td></tr> <tr> <td>ControlOperator</td><td>SysML::Activities::C</td></tr> </tbody> </table> <p>Applied Stereotypes:</p> <ul style="list-style-type: none"> <li>TestCase</li> </ul>	Stereotype	Information	Allocated	SysML::Allocations::	Block	SysML::Blocks::Bloc	ConstraintBlock	SysML::Constraints::	ControlOperator	SysML::Activities::C	<pre> &gt; &lt;Model&gt; RootElement   &gt; &lt;TestCase&gt; Activity1     &lt;Stereotype&gt; TestCase [stereotype applied]           </pre>
Stereotype	Information										
Allocated	SysML::Allocations::										
Block	SysML::Blocks::Bloc										
ConstraintBlock	SysML::Constraints::										
ControlOperator	SysML::Activities::C										

...

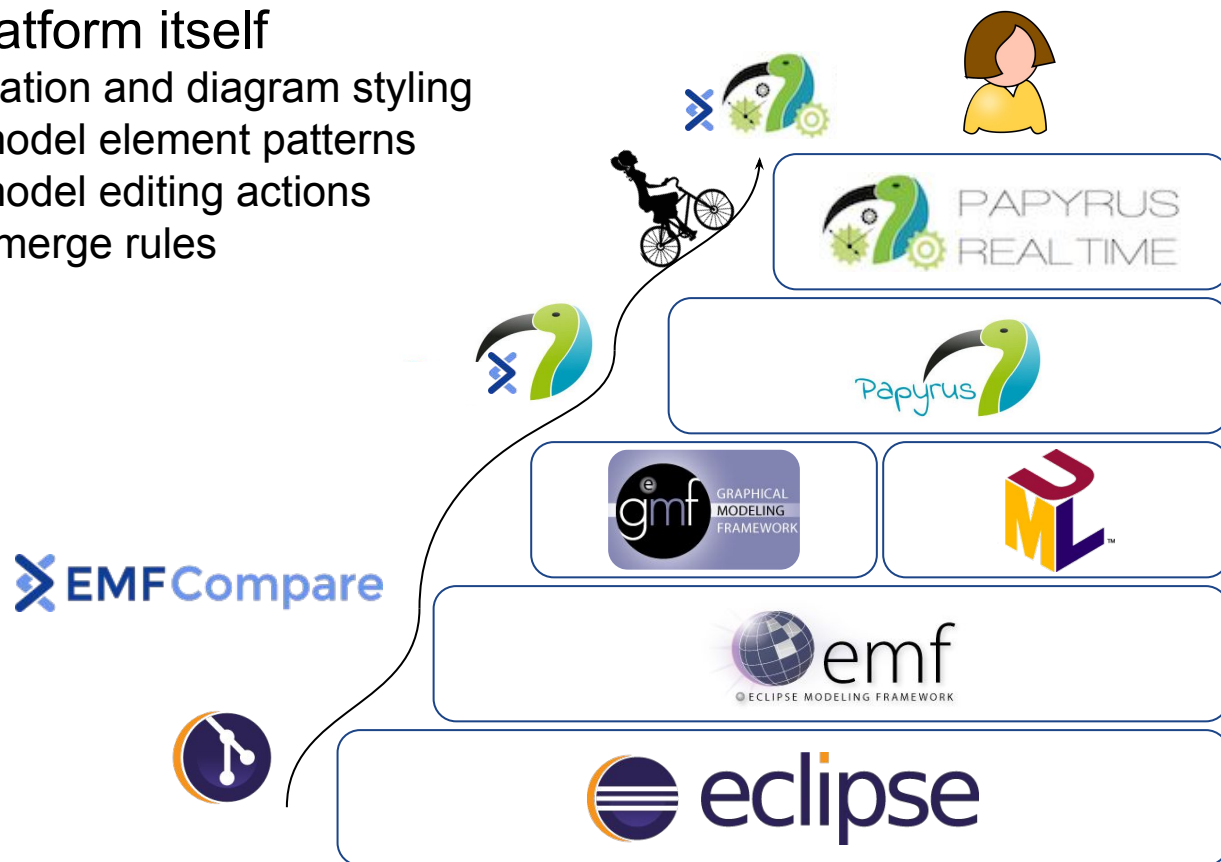
## “Papyrus Compare”

- Seamless Papyrus integration
  - ✓ Direct integration of Papyrus’ tree decorations
  - ✓ Integration of Papyrus diagram styling
  - ✓ Support for UML Profiles
  - ✓ Support for UML Profile migration
  - ✓ Grouping of several composite editing actions
  - ✓ Specific conflict handling for those actions
  - ✓ Specific handling of metainformation (di)
  - ✓ ...

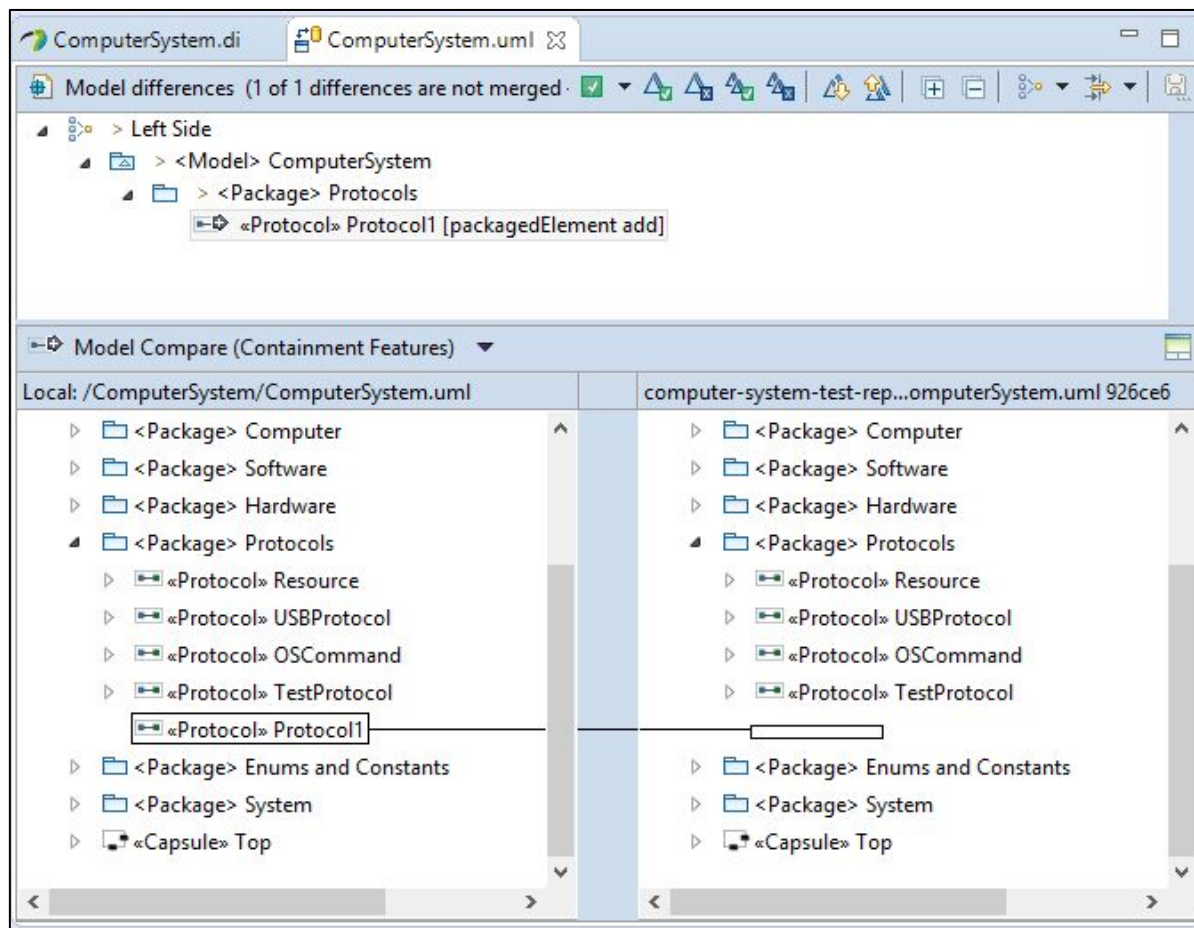


## Gap between Model Versioning Tools and **DSM** Editors

- Papyrus is a DSM platform itself
  - Custom model decoration and diagram styling
  - Custom composite model element patterns
  - Custom composite model editing actions
  - Custom conflict and merge rules
  - ...



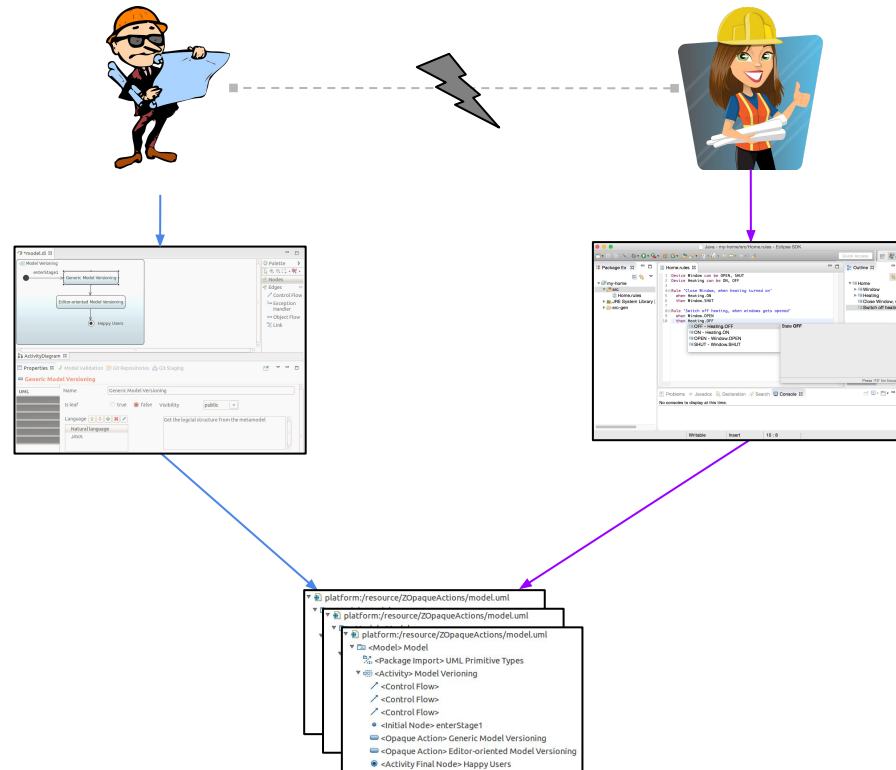
## Gap between Model Versioning Tools and Modeling Editors



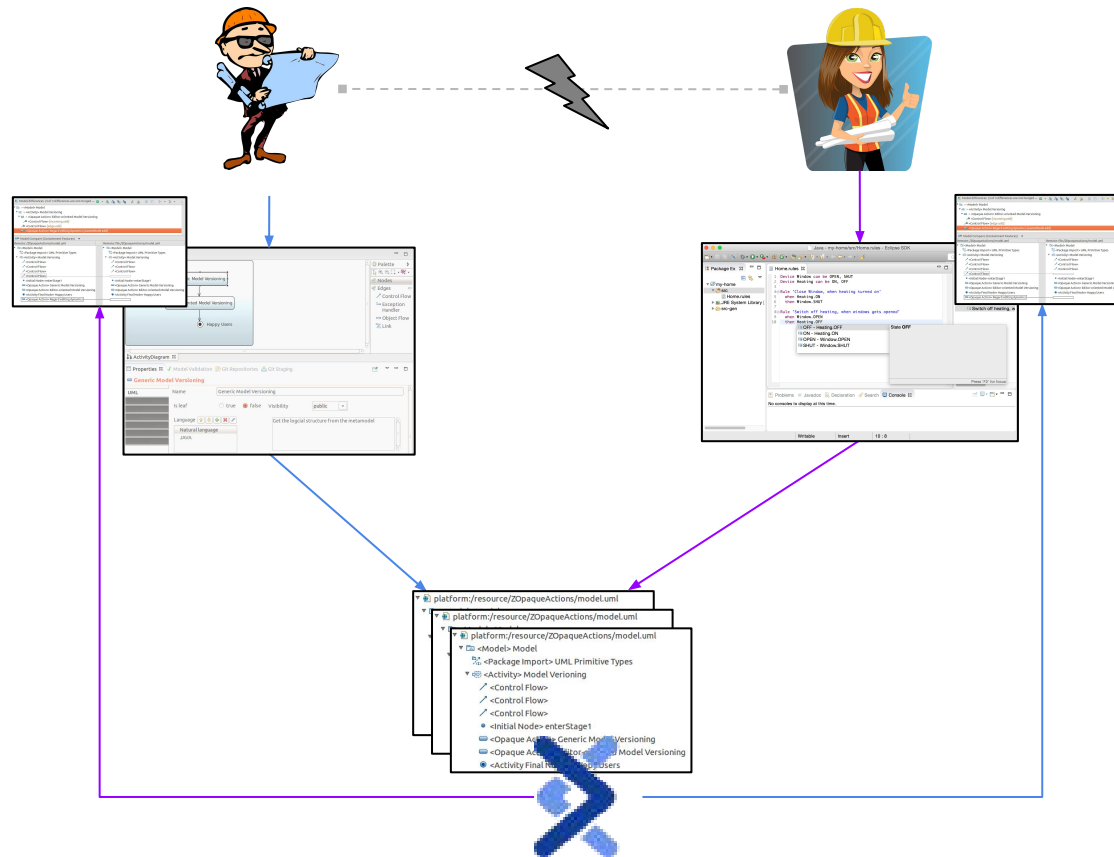
## Outlook

- Many customization options exist in Papyrus
  - Reflect all customizations in the model versioning process too
  - Seamless UX: See model changes on the same level as in the editor
- Shared DSML customization API
  - Perform a customization of a DSM tool once
  - See the effect of the customization everywhere
- From DSML-oriented to “representation”-oriented model versioning
  - Different user roles work on shared models
  - They usually look at the models at different levels
  - Consequently, they use different representations: tables, diagrams, text
  - Allow them to view changes in *their* preferred representation

# Communicate Changes & Resolve Conflicts Across Multiple Domain-specific Representations



# Communicate Changes & Resolve Conflicts Across Multiple Domain-specific Representations





# Thank you very much!

- Open source collaboration 



- Follow us on [collaborative-modeling.org](http://collaborative-modeling.org)



- Contact us
  - At our booth (EclipseSource, Zeligsoft, PapyrusIC)
  - [planger@eclipsesource.com](mailto:planger@eclipsesource.com)

## Related Sessions

Today 13:45  
Papyrus Customization and  
DSML support  
Benoit Maggi (CEA)

Today 15:30  
Collaborative Modeling  
P. Langer & M. Koegel